

☐ Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d)) for the total number of months checked below:

| <u>Months</u> | <u>Large Entity</u> | <u>Small Entity</u> |
|--------------------------------|---------------------|---------------------|
| <input type="checkbox"/> one | \$120.00 | \$60.00 |
| <input type="checkbox"/> two | \$450.00 | \$225.00 |
| <input type="checkbox"/> three | \$1,020.00 | \$510.00 |
| <input type="checkbox"/> four | \$1,590.00 | \$795.00 |

If an additional extension of time is required, please consider this a petition therefor.

☐ An extension for __ months has already been secured and the fee paid therefore of \$ is deducted from the total fee due for the total months of extension now requested.

☒ Applicant believes that no extension of time is required. However, this conditional petition is being made to provide for the possibility that Applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Total Fees Due:

| | |
|------------------------|------------------------|
| Appeal Brief Fee | \$500.00 |
| Extension Fee (if any) | \$ |
| Total Fee Due | <u>\$500.00</u> |


☒ Enclosed is Check No. 17656 in the amount of \$500.00.

☐ The Commissioner is authorized to charge the total fees due of \$____ to Deposit Account No. 50-0850, (Order No. ____).

☒ The Commissioner is authorized to charge any additional required fees or credit any overpayment to Deposit Account No. 50-0850, (Order No. SUNMP024).

One additional copy of this transmittal is enclosed for fee processing.

Respectfully submitted,
MARTINE PENILLA & GENCARELLA, LLP


Kenneth D. Wright
Reg. No. 53,795

710 Lakeway Drive, Suite 200
Sunnyvale, CA 94085
(408) 749-6900
Customer No. 32,291



PATENT

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

EX PARTE GUPTA

Application for Patent

Filed October 23, 2001

Application No. 10/038,338

FOR:

XML Based Report Generator

APPEAL BRIEF

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on January 16, 2007.

Signed: _____


Kenneth D. Wright

MARTINE PENILLA & GENCARELLA, LLP
Attorneys for Applicant

01 FC:1402

533.03 0P

TABLE OF CONTENTS

| | <u>Page No.</u> |
|---|------------------------|
| I. REAL PARTY IN INTEREST | 3 |
| II. RELATED APPEALS AND INTERFERENCES | 3 |
| III. STATUS OF THE CLAIMS | 3 |
| IV. STATUS OF THE AMENDMENTS | 3 |
| V. SUMMARY OF CLAIMED SUBJECT MATTER | 3 |
| VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL | 7 |
| VII. ARGUMENTS | 7 |
| A. Rejections of Claims 1-12, 14-17, and 19-21 under 35 U.S.C. 103(a) | 7 |
| VIII. CLAIMS APPENDIX | 21 |
| IX. EVIDENCE APPENDIX | 26 |
| X. RELATED PROCEEDINGS APPENDIX | 26 |

I. REAL PARTY IN INTEREST

The real party in interest is Sun Microsystems, Inc., the assignee of the present application.

II. RELATED APPEALS AND INTERFERENCES

The Applicant is not aware of any related appeals or interferences.

III. STATUS OF THE CLAIMS

A total of 21 claims were presented during prosecution of this application. Claims 13 and 18 were cancelled. The Applicant appeals rejected claims 1-12, 14-17, and 19-21.

IV. STATUS OF THE AMENDMENTS

Claims 1, 6, 8, 11-12, 14-16, and 21 were amended prior to the Final Office Action from which this Appeal is made. A response to the Final Office Action of July 12, 2006, was filed on September 12, 2006, and did not include further claim amendments. The Examiner did not consider the Applicant's arguments in the response of September 12, 2006, to be persuasive.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 recites one embodiment of the present invention directed to a method for creating a test summary report. The method includes an operation for executing a computer software test application on a platform (p. 10, lines 22-23) The computer software test application has one or more test suites (p. 10, line 23, through p. 11, line 1). Each test suite includes one or more test cases (p. 8, lines 9-10). Each test case is configured to test an aspect of the platform (p. 10, line 23, through p. 11, line 1). The method also includes an operation for generating test results in an Extensible Markup

Language (XML) enabled format that is independent of an XML format capability of the platform (p. 11, lines 4-15). The method further includes an operation for using the XML enabled test results to create a test summary report (p. 11, lines 18-19). The XML enabled test results are capable of being rearranged (p. 12, line 12, through p. 13, line 7). The rearranged XML enabled test results include test suite tags (p. 15, lines 7-14). Each test suite tag encapsulates the test results corresponding to each test suite of the computer software test application (p. 15, lines 10-13).

Independent claim 11 recites one embodiment of the present invention directed to an Extensible Markup Language (XML) based report generator. The XML based report generator includes a parser that processes a test execution log file to generate a well-formed XML test reports file (p. 12, lines 12-15). The test execution log file includes XML statements generated independent of an XML format capability of a platform being tested (p. 11, lines 4-15). The XML based report generator also includes a logical parser that can be operated to use the well-formed XML test reports file to reorder a content of the well-formed XML test reports file so as to produce a logically arranged XML test reports file (p. 14, lines 1-3). The logically arranged XML test reports file includes test suite tags that encapsulate the test results corresponding to each test suite (p. 15, lines 7-14). The XML based report generator further includes an HTML converter parser that converts the logically arranged XML test reports file into an HTML test summary file (p. 15, lines 15-17).

Independent claim 16 recites one embodiment of the present invention directed to a method for creating a test summary report. The method includes an operation for executing a test application on a platform (p. 10, lines 22-23). The test application is executed using a status utility having functions that generate XML code (p. 11, lines 6-13). The test application has one or more test suites (p. 10, line 23, through p. 11, line

1). Each test suite has one or more test cases (p. 8, lines 9-10). Each test case is configured to test an aspect of the platform (p. 10, line 23, through p. 11, line 1). The method also includes an operation for generating test results in an Extensible Markup Language (XML) enabled format that is independent of an XML format capability of the platform using the status utility (p. 11, lines 4-15). The test results are output to a test execution log file (p. 11, lines 10-13). The method further includes an operation for using the test execution log file to generate a well-formed XML test reports file (p. 12, lines 12-15). An operation is performed to reorder a content of the well-formed XML test reports file to create a logically arranged XML test reports file, wherein test suite tags are generated for each test suite (p. 14, lines 1-3). The test suite tags encapsulate the test results corresponding to each test suite (p. 15, lines 7-14). The method also includes an operation for converting the logically arranged XML test reports file into an HTML test summary report (p. 15, lines 15-17).

Independent claim 21 recites one embodiment of the present invention directed to a computer generated method of generating and processing test results. The method includes an operation for executing one or more tests from a plurality of test suites (p. 17, lines 10-14). A test execution log file is generated, wherein each test result includes data identifying the test to which the test result relates and the test suite to which the test belongs (p. 17, lines 10-14, and p. 14, lines 10-13). Each test suite of the plurality of test suites has one or more test cases (p. 8, lines 9-10). Each test case is configured to test an aspect of the platform (p. 10, line 23, through p. 11, line 1). The method also includes an operation for processing the test execution log file to generate a well-formed XML based test reports file that is independent of an XML format capability of a platform being tested (p. 12, lines 12-15). The test results in the test reports file are arranged as a plurality of independent test results (p. 14, lines 13-15). Each test result in the test reports file

includes a test ID and data identifying a test suite to which a test belongs (p. 14, lines 13-15). The method further includes processing the well-formed XML test reports file to logically arrange the well-formed XML test reports file so as to create a logically arranged XML test reports file (p. 15, lines 7-10). In the logically arranged XML test reports file, tags are generated for each test suite and the tags encapsulate the test results corresponding to each test suite (p. 15, lines 10-13). Additionally, the method includes an operation for outputting the logically arranged well-formed XML test reports file (p. 15, lines 13-14).

It should be appreciated that the above discussion represents only a summary of the present invention. A more in-depth discussion of the present invention is provided in the Detailed Description section of the application.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1-12, 14-17, and 19-21 were rejected under 35 U.S.C. 103(a) as being unpatentable over Mason (U.S. Patent No. 6,826,716) in view of Davis (U.S. Patent Application Publication No. US2005/0198042A1) in view of Washburn et al. ("Washburn" hereafter) (U.S. Patent No. 5,157,779).

VII. ARGUMENTS

A. Rejections of Claims 1-12, 14-17, and 19-21 under 35 U.S.C. 103(a)

Independent Claim 1

Claim 1 was rejected under 35 U.S.C. 103(a) as being unpatentable over Mason in view of Davis in further view of Washburn. The following clear errors in the Examiner's rejection are noted.

1. **The combination of Mason, Davis, and Washburn does not teach generating test results in an Extensible Markup Language (XML) enabled format independent of an XML format capability of the platform.**

In the Final Office Action of July 12, 2006, page 3, the Examiner asserts that Mason (column 15, lines 25-35) teaches the following features of claim 1: "generating test results in an Extensible Markup Language (XML) enabled format independent of an XML format capability of the platform."

Mason (column 15, lines 25-35) actually states the following:

"In light of the detailed description set forth, readers will observe that advantages of the use of typical embodiments of the present invention included that a test

generator in the J2EE environment can have detailed knowledge of the server side quality of service attributes expected of the enterprise application via the XML deployment descriptors, and a test generator could generate tests to explicitly test such attributes. It can potentially generate many different kinds of test each targeted at a specific aspect of quality or verification. For instance, correctness, load/stress test, reliability, long running tests, and so on."

Mason's teachings are associated with a test generator that is defined to generate tests for an enterprise application. Mason (column 15, lines 25-35) teaches that a test generator operating in the J2EE environment is capable of having detailed knowledge of server side quality of service (QoS) attributes expected of an enterprise application. Specifically, Mason (column 15, lines 25-35) teaches that the test generator obtains the detailed knowledge of the server side QoS attributes expected of the enterprise application by analyzing XML deployment descriptors associated with deployment of the enterprise application. Mason (column 2, lines 3-6) teaches that the XML deployment descriptors are text files that specify component, i.e., enterprise application, behavior in terms of well-defined XML tags. Using the XML deployment descriptors, attributes of the enterprise application can be configured at the time of its deployment to comply with specific environmental requirements.

It should be understood that Mason (column 2, lines 10-20) simply defines quality of service to be application behaviors or XML tags related to the security or transactional control aspects of an enterprise application. Additionally, Mason (column 2, lines 10-20) defines an XML parser as a software tool capable of reading an XML document and breaking down the XML elements into usable parts.

In accordance with the foregoing, Mason's teachings (column 15, lines 25-35) are related to the test generator's ability to generate tests for an enterprise application that are

explicitly related to the server side QoS attributes expected of the enterprise application. Specifically, Mason teaches that the XML deployment descriptors associated with deployment of the enterprise application are analyzed by the test generator to determine which attributes are exercised by the XML deployment descriptors when the enterprise application is deployed. Then, Mason teaches that with the knowledge gleaned from the analysis of the XML deployment descriptors, the test generator is able to generate tests for the enterprise application that are specifically related to the particular attributes referenced within the XML deployment descriptors, thereby tailoring the generated tests to the particular attributes of the enterprise application.

It should be understood that the XML deployment descriptors discussed by Mason do not represent test results. Rather, the XML deployment descriptors of Mason facilitate deployment of the enterprise application, and thereby include information about the QoS attributes of the enterprise application. The information about the QoS attributes of the enterprise application, as obtained through analysis of the XML deployment descriptors, are used by a test generator to generate a software test that is specifically tailored to verify the QoS attributes of the enterprise application. Simply stated, the XML deployment descriptors of Mason do not represent the result of a software test. Rather, the XML deployment descriptors of Mason are used as an input to generate a software test.

In rejecting claim 1, the Examiner has relied solely upon Mason (within the combination of Mason, Davis, and Washburn) to teach the feature of "generating test results in an Extensible Markup Language (XML) enabled format independent of an XML format capability of the platform." In contrast to the Examiner's assertions, the teachings of Mason, particularly those teachings associated with XML deployment descriptors, are not descriptive of test results. Specifically, the teachings of Mason, particularly those teachings

associated with XML deployment descriptors, do not teach generation of test results in an XML enabled format independent of an XML format capability of the platform.

Furthermore, claim 1 requires that the XML enabled format in which the test results are generated be independent of an XML format capability of the platform. Although the Examiner has asserted that Mason teaches generation of test result in an XML enabled format, the Examiner has not cited a teaching in Mason regarding the requirement that the XML enabled format be independent of the XML format capability of the platform. However, in discussing the teachings of Davis, the Examiner mentions the feature of claim 1 regarding the XML enabled format of the generated test results being independent of the XML format capability of the platform. However, the Examiner does not clearly communicate how Davis is interpreted to teach the XML enabled format of the generated test results being independent of the XML format capability of the platform. Additionally, the Examiner has failed to provide a reasoned explanation as to how the disclosure of Davis is thought to be combinable with the disclosure of Mason to teach the feature of claim 1 requiring the generation of test results in the XML enabled format that is independent of an XML format capability of the platform.

In view of the foregoing, the Applicant submits that the teachings of Mason, particularly those provided at (column 15, lines 25-35) and (column 2, lines 10-20), do not teach generating test results in an XML enabled format that is independent of an XML format capability of the platform, as recited in claim 1. Moreover, the cited teachings of Mason are not relevant to a format of test results, much less the XML enabled format that is independent of an XML format capability of the platform, as recited in claim 1.

To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). “All words in a claim must be considered in judging the

patentability of that claim against the prior art.” *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970). Based at least on the foregoing discussion of how the disclosure of Mason fails to teach the features of claim 1, the Applicant submits that all the limitations of claim 1 are not taught or suggested by the combined prior art. Therefore, the Applicant submits that the Examiner's rejection of claim 1 under 35 U.S.C. 103 is in error. The Board is respectfully requested to overturn the Examiner's rejection of claim 1 under 35 U.S.C. 103.

2. **The combination of Mason, Davis, and Washburn does not teach using the XML enabled test results to create a test summary report.**

In the Final Office Action, page 3, the Examiner asserts that Davis (abstract) teaches the following features of claim 1: "using the XML enabled test results to create a test summary report."

Davis (abstract) actually states the following:

"Methods and systems provide a 'chart view' for a markup language referred to as Reusable Data Markup Language ('RDML'). Generally, a chart view comprises the components necessary for automatically manipulating and displaying a graphical display of numerical data contained in RDML markup documents. RDML is a markup language, such as the Hypertext Markup Language ('HTML') or the Extensible Markup Language ('XML')."

Davis teaches a method for providing a "chart view" for a markup language. The chart view of Davis includes the components necessary for automatically manipulating and graphically displaying numerical data contained in a reusable data markup language (RDML), wherein RDML is a markup language such as HTML or XML. The chart view of

Davis transforms, formats, manipulates, and displays data stored in the RDML markup documents using attributes that describe the meaning of the data.

In applying the combination of Mason, Davis, and Washburn to reject claim 1, the Examiner has cited Davis as teaching the operation for using the XML enabled test results to create a test summary report, as recited in claim 1. The Examiner asserts that the chart view method of Davis teaches a mechanism by which data in an XML document can be transformed, formatted, manipulated, and displayed based on attributes associated with the data in the XML document. The Examiner further asserts that the chart view method of Davis is capable of being used to perform the operation of claim 1 regarding using the XML enabled test results to create a test summary report.

While the chart view of Davis may be considered to have the CAPABILITY of using XML based test results to create a test summary report, Davis does not actually teach an operation for using XML based test results to create a test summary report. In fact, Davis is silent with regard to XML based test results and the creation of a test summary report using XML based test results.

To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). “All words in a claim must be considered in judging the patentability of that claim against the prior art.” *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970). Based at least on the foregoing discussion of how the disclosure of Davis fails to teach the features of claim 1, the Applicant submits that all the limitations of claim 1 are not taught or suggested by the combined prior art. Therefore, the Applicant submits that the Examiner's rejection of claim 1 under 35 U.S.C. 103 is in error. The Board is respectfully requested to overturn the Examiner's rejection of claim 1 under 35 U.S.C. 103.

Furthermore, the Applicant submits that it is only possible through the use of hindsight as afforded by the disclosure of the present invention to construe Davis as teaching the claimed feature of using XML based test results to create a test summary report. Per MPEP 2142, impermissible hindsight must be avoided when considering whether a claim is prima facie obvious under 35 U.S.C. 103, and the legal conclusion must be reached on the basis of the facts gleaned from the prior art, not from the Applicant's disclosure. Because Davis is silent with regard to XML based test results and a test summary report created therefrom, the Applicant submits that the Examiner's assertions regarding the teachings of Davis are based on the Applicant's disclosure rather than facts gleaned from the prior art.

3. **The combination of Mason, Davis, and Washburn does not teach test suite tags included within XML enabled test results.**

In the Final Office Action, page 6, the Examiner asserts that Washburn (column 1, line 50, through column 2, line 15, and Figures 2 and 11b) teaches the following features of claim 1: "the rearranged XML enabled test results including test suite tags, each test suite tag encapsulating the test results corresponding to each test suite of the computer software test application."

In applying the combination of Mason, Davis, and Washburn to reject claim 1, the Examiner has relied upon Washburn to teach the test suite tags included within the XML enabled test results. Washburn, however, is silent with regard to XML and more particularly with regard to XML enabled test results. Also, as discussed above Mason does not teach XML enabled test results either. Therefore, the Applicant reiterates that the reference combination does not teach XML enabled test results.

In the history of XML, XML 1.0 became a World Wide Web Consortium (W3C) recommendation on February 10, 1998. The Washburn reference was filed as a patent application on June 7, 1990. Therefore, the Washburn reference predates the advent of XML by more than seven years. Thus, the Applicant submits that it is not reasonable for the Examiner to even attempt to construe Washburn as teaching a type of XML tag, much less the test suite tag included in the XML enabled test results as recited in claim 1. As the reference combination of Mason, Davis, and Washburn fails to teach the test suite tag of claim 1, it follows that the reference combination also fails to teach the test suite tag property of encapsulating the test results corresponding to each test suite of the computer software test application.

To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). "All words in a claim must be considered in judging the patentability of that claim against the prior art." *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970). Because the combination of Mason, Davis, and Washburn fails to teach the test suite tags included within the XML enabled test results, the Applicant submits that all the limitations of claim 1 are not taught or suggested by the combined prior art. Therefore, the Applicant submits that the Examiner's rejection of claim 1 under 35 U.S.C. 103 is in error. The Board is respectfully requested to overturn the Examiner's rejection of claim 1 under 35 U.S.C. 103.

4. **There is no motivation found within either Mason, Davis, or Washburn that would have led one skilled in the art at the time of the invention to combine their respective teachings in the manner asserted by the Examiner.**

Notwithstanding the fact that the combination of Mason, Davis, and Washburn fails to teach each and every feature of claim 1, the Applicant submits that there is no motivation or suggestion within the Mason, Davis, and Washburn references to have combined their teachings in the manner suggested by the Examiner. Specifically, as discussed above, Mason is not concerned with (and does not teach) generating test results in an XML enabled format. Also, as discussed above, Davis is not concerned with (and does not teach) generating test results in an XML enabled format and using the XML enabled test results to create a test summary report. Furthermore, Washburn is not concerned with (and does not teach) anything related to XML, much less generating test results in an XML enabled format. Therefore, the Applicant submits that the Examiner has effectively and inappropriately used the Applicant's disclosure as a blueprint for piecing together unrelated teachings of the prior art in an attempt to establish a prima facie case of obviousness against claim 1. Simply stated, the motivation and/or suggestion to combine the Mason, Davis, and Washburn references in the particular manner suggested by the Examiner is not found within the references themselves.

The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on applicant's disclosure. *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). Furthermore, the level of skill in the art cannot be relied upon to provide the suggestion to combine references. *Al-Site Corp. v. VSI Int'l Inc.*, 174 F.3d 1308, 50 USPQ2d 1161 (Fed. Cir. 1999). Also, the mere fact that references can be combined or modified does not render the resultant combination obvious unless the prior art also suggests the desirability of the combination. *In re Mills*, 916 F.2d 680, 16 USPQ2d 1430 (Fed. Cir. 1990) Furthermore, impermissible hindsight based upon the Applicant's disclosure must be avoided (MPEP

2142). The legal conclusion of the Examiner regarding obviousness under 35 U.S.C. 103 must be reached on the basis of the facts gleaned from the prior art (MPEP 2142).

In view of the foregoing, the Applicant submits that the Examiner's rejection of claim 1 under 35 U.S.C. 103 is in error. The Board is respectfully requested to overturn the Examiner's rejection of claim 1 under 35 U.S.C. 103.

Summary of Arguments with Regard to Claim 1

The Examiner's rejections are inappropriate for at least the following reasons:

- Argument 1: The combination of Mason, Davis, and Washburn does not teach generating test results in an Extensible Markup Language (XML) enabled format independent of an XML format capability of the platform.
- Argument 2: The combination of Mason, Davis, and Washburn does not teach using the XML enabled test results to create a test summary report.
- Argument 3: The combination of Mason, Davis, and Washburn does not teach test suite tags included within XML enabled test results.
- Argument 4: There is no motivation found within either Mason, Davis, or Washburn that would have led one skilled in the art at the time of the invention to combine their respective teachings in the manner asserted by the Examiner.

Independent Claim 11

Claim 11 was rejected under 35 U.S.C. 103(a) as being unpatentable over Mason in view of Davis in further view of Washburn. The following clear errors in the Examiner's rejection are noted.

The combination of Mason, Davis, and Washburn does not teach a parser that processes a test execution log file to generate a well-formed XML test reports file,

wherein the test execution log file includes XML statements generated independent of an XML format capability of a platform being tested.

In the Final Office Action, page 9, the Examiner asserts that Mason (column 1, line 65, through column 2, line 5) teaches the following features of claim 1: "a parser that processes a test execution log file to generate a well-formed XML test reports file." Also, in the Final Office Action, page 10, the Examiner asserts that Davis teaches that "the test execution log file includes XML statements generated independent of an XML format capability of a platform being tested."

Mason (column 1, line 65, through column 2, line 7) states the following:

"Central to the J2EE component-based development model is the notion of containers. Containers are standardized runtime environments that provide specific component services. In addition, containers provide a mechanism for selecting application behaviors at assembly or deployment time. Through the use of deployment descriptors (text files that specify component behavior in terms of well-defined XML tags), components can be configured to a specific container's environment when deployed, rather than in component code."

The Applicant submits that neither the containers, the deployment descriptors, nor the components, as referenced in the above-cited portion of Mason, teach a parser that processes a test execution log file to generate a well-formed XML test reports file. The Examiner has not communicated how the disclosure of Mason (column 1, line 65, through column 2, line 7) is interpreted to teach the parser that processes a test execution log file to generate a well-formed XML test reports file, as recited in claim 1. Also, the Examiner does not clearly communicate how Davis is interpreted to teach that the test execution log file includes XML statements generated independent of an XML format capability of a platform being tested.

Additionally, the Examiner has asserted that claim 11 includes subject matter similar to that recited in claim 1. The Examiner has also asserted that features of claim 11 that are similar to features of claim 1 are rejected for the same reasons as discussed with regard to claim 1. Therefore, the Applicant submits that the arguments presented above with regard to claim 1 are equally applicable to similar features recited in claim 11.

To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). “All words in a claim must be considered in judging the patentability of that claim against the prior art.” *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970). Because the combination of Mason, Davis, and Washburn fails to teach a parser that processes a test execution log file to generate a well-formed XML test reports file, wherein the test execution log file includes XML statements generated independent of an XML format capability of a platform being tested, the Applicant submits that all the limitations of claim 11 are not taught or suggested by the combined prior art. Therefore, the Applicant submits that the Examiner's rejection of claim 11 under 35 U.S.C. 103 is in error. The Board is respectfully requested to overturn the Examiner's rejection of claim 1 under 35 U.S.C. 103.

Independent Claims 16 and 21

With respect to independent claims 16 and 21, the Examiner has used the same bases of rejection as applied to similar features recited in claims 1 and 11. Therefore, the Applicant submits that the arguments presented above for claims 1 and 11 are equally applicable to each of claims 16 and 21. Thus, the Applicant submits that the combination of Mason, Davis, and Washburn fails to teach each and every feature of claims 16 and 21, respectively, as required to render each of claims 16 and 21 *prima facie* obviousness under

35 U.S.C. 103. Therefore, the Applicant respectfully submits that the Examiner's rejections of claims 16 and 21 under 35 U.S.C. 103 are in error. The Board is respectfully requested to overturn the Examiner's rejection of claims 16 and 21 under 35 U.S.C. 103.

Dependent Claims 2-10, 12, 14-15, 17, and 19-20

Because a dependent claim incorporates each and every feature of its independent claim, the Applicant submits that each of dependent claims 2-10, 12, 14-15, 17, and 19-20 is patentable with respect to the cited art of record for at least the same reasons provided for its respective independent claim.

Conclusion

In view of the foregoing, the Applicant submits that the combination of Mason, Davis, and Washburn fails to teach each and every feature of each of independent claims 1, 11, 16, and 21, as required to establish prima facie obviousness under 35 U.S.C. 103. To establish prima facie obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). Also, with regard to each of independent claims 1, 11, 16, and 21, the Applicant submits that there is no motivation found within either Mason, Davis, or Washburn, as required by 35 U.S.C. 103, that would have led one skilled in the art to combine their respective teachings in the manner asserted by the Examiner.

In view of the foregoing, the Applicant submits that each of claims 1-12, 14-17, and 19-21 is patentable. Therefore, the Applicant respectfully requests that the Board of Patent Appeals and Interferences reverse the Examiner's rejections of the claims on appeal.

Respectfully Submitted,
MARTINE PENILLA & GENCARELLA, LLP

A handwritten signature in black ink, appearing to read 'Kenneth D. Wright', is written over the printed name.

Kenneth D. Wright
Reg. No. 53,795

MARTINE PENILLA & GENCARELLA, LLP
710 Lakeway Drive, Suite 200
Sunnyvale, California 94085
408.749.6900

VIII. CLAIMS APPENDIX

1. A method for creating a test summary report, comprising the operations of:

executing a computer software test application on a platform, the computer software test application having one or more test suites, each test suit including one or more test cases, each test case configured to test an aspect of the platform;

generating test results in an Extensible Markup Language (XML) enabled format independent of an XML format capability of the platform; and

using the XML enabled test results to create a test summary report,

wherein the XML enabled test results is capable of being rearranged, the rearranged XML enabled test results including test suite tags, each test suite tag encapsulating the test results corresponding to each test suite of the computer software test application.

2. A method as recited in claim 1, wherein the test results are generated utilizing a status utility having functions that generate XML code.

3. A method as recited in claim 1, wherein the test results are output to a test execution log file, the test execution log file including a log of the test execution.

4. A method as recited in claim 3, further including the operation of processing the test execution log file to generate a well-formed XML test reports file.

5. A method as recited in claim 4, wherein the well-formed XML test reports file is further valid with respect to a Test document type definition (DTD).

6. A method as recited in claim 4, wherein a content of the well-formed XML test reports file is reordered so as to create a logically arranged XML test reports file.

7. A method as recited in claim 6, wherein the logically arranged XML test reports file includes test suite tags indicating test reports that belong to particular test suites.

8. A method as recited in claim 6, wherein the test summary report is generated by converting the logically arranged XML test reports file into a Hypertext Markup Language (HTML) test summary report.

9. A method as recited in claim 8, wherein the HTML test summary report provides a test summary of the test execution log file.

10. A method as recited in claim 9, wherein the HTML test summary report includes links to failure description pages, wherein the failure description pages provide a detailed description of a particular test failure.

11. An Extensible Markup Language (XML) based report generator, comprising:

a parser that processes a test execution log file to generate a well-formed XML test reports file, wherein the test execution log file includes XML statements generated independent of an XML format capability of a platform being tested;

a logical parser operable to use the well-formed XML test reports file to reorder a content of the well-formed XML test reports file to produce a logically arranged XML test reports file, the logically arranged XML test reports file including test suite tags encapsulating the test results corresponding to each test suite; and

an HTML converter parser that converts the logically arranged XML test reports file into an HTML test summary file.

12. An XML based report generator as recited in claim 11, wherein the well-formed XML test reports file is valid with respect to a Test document type definition (DTD).

14. An XML based report generator as recited in claim 11, wherein the HTML test summary report provides a test summary of the test execution log file.

15. An XML based report generator as recited in claim 14, wherein the HTML test summary report includes links to failure description pages, wherein the failure description pages provide a detailed description of a particular test failure.

16. A method for creating a test summary report, comprising the operations of:

executing a test application on a platform, wherein the test application is executed using a status utility having functions that generate XML code, the test application having one or more test suites, each test suite having one or more test cases, each test case configured to test an aspect of the platform;

generating test results in an Extensible Markup Language (XML) enabled

format independent of an XML format capability of the platform using the status utility, wherein the test results are output to a test execution log file;

using the test execution log file to generate a well-formed XML test reports file;

reordering a content of the well-formed XML test reports file to create a logically arranged XML test reports file wherein test suite tags are generated for each test suite, each test suite tag encapsulating the test results corresponding to each test suite; and

converting the logically arranged XML test reports file into an HTML test summary report.

17. A method as recited in claim 16, wherein the well-formed XML test reports file is further valid with respect to a Test document type definition (DTD).

19. A method as recited in claim 16, wherein the HTML test summary report provides a test summary of the test execution log file.

20. A method as recited in claim 19, wherein the HTML test summary report includes links to failure description pages, wherein the failure description pages provide a detailed description of a particular test failure.

21. A computer generated method of generating and processing test results, comprising the operations of:

executing one or more tests from a plurality of test suites and generating test execution log file wherein each test result includes data identifying the test to which the test result relates the test suite to which the test belongs, each test suite of the plurality of

test suites having one or more test cases, each test case configured to test an aspect of the platform;

processing the test execution log file to generate a well-formed XML based test reports file independent of an XML format capability of a platform being tested, wherein the test results in the test reports file are arranged as a plurality of independent test results, each including a test ID and data identifying a test suite to which a test belongs;

processing the well-formed XML test reports file to logically arrange the well-formed XML test reports file to create a logically arranged XML test reports file in which tags are generated for each test suite the tags encapsulating the test results corresponding to each test suite; and

outputting the logically arranged well-formed XML test reports file.

IX. EVIDENCE APPENDIX

There is currently no evidence entered and relied upon in this Appeal.

X. RELATED PROCEEDINGS APPENDIX

There are currently no decisions rendered by a court or the Board in any proceeding identified in the Related Appeals and Interferences section.